CLAIMS

What is claimed is:

1. A method providing partial speculative operation in lieu of suspending speculation, said method comprising:

operating in a first mode of speculative operation, said first mode permitting speculation of a first set of speculative operations; and

exiting said first mode and entering a second mode of speculative operation in response to an event, said second mode permitting speculation of a second set of speculative operations that is a subset of said first set.

2. The method of Claim 1 wherein said first set of speculative operations comprises microprocessor register operations, operations that involve memory that is private to a microprocessor, input/output (I/O) writes, main memory reads, main memory writes, and non-architectural faults.

3. The method of Claim 1 wherein said second set of speculative operations comprises microprocessor register operations, operations that involve memory that is private to a microprocessor, and architectural faults.

4. The method of Claim 1 wherein said second set of speculative operations comprises speculative operations that are invisible external to a microprocessor.

5. The method of Claim 1 wherein said event is selected from the group consisting of a fault, a direct memory access request, and an I/O read.

6. The method of Claim 1 further comprising suspending speculative operation in response to a second event.

7. The method of Claim 1 further comprising returning to said first mode after said event is handled.

8. The method of Claim 1 further comprising:

counting the number of instructions executed in said first mode prior to said event; and

returning to said first mode upon executing the same number of instructions after entering said second mode.

9. The method of Claim 1 implemented using a microprocessor comprising a combination of translation software and host hardware, said translation software running directly on said host hardware, said translation software for interpreting and translating a sequence of non-native instructions into a sequence of native instructions, wherein said interpreting is permitted during said second mode.

10. A method providing partial speculative operation, said method comprising:

executing forward from a speculation boundary representing a memory state, said executing according to a full speculation mode that permits a set of speculative operations;

experiencing an event during said executing;

rolling back to said speculation boundary and restoring said memory state in response to said event;

executing forward from said speculation boundary according to a partial speculation mode that permits a subset of said set of speculative operations, said partial speculation mode used in lieu of suspending said set of speculative operations in entirety.

11. The method of Claim 10 wherein said set of speculative operations comprises microprocessor register operations, operations that involve memory that is private to a microprocessor, input/output (I/O) writes, main memory reads, main memory writes, and non-architectural faults.

12. The method of Claim 10 wherein said subset of speculative operations comprises microprocessor register operations, operations that involve memory that is private to a microprocessor, and architectural faults.

13. The method of Claim 10 wherein said subset of speculative operations comprises speculative memory operations that are invisible external to a microprocessor.

14. The method of Claim 10 wherein said event is selected from the group consisting of a fault, a direct memory access request, and an I/O read.

15. The method of Claim 10 further comprising:

detecting a second event during operation in said partial speculation mode; and

suspending speculative operation in response to said second event.

16. The method of Claim 10 further comprising:

handling said event; and

returning to said full speculation mode after said event is handled.

17. The method of Claim 10 further comprising:

counting the number of instructions executed in said full speculation mode prior to said event;

executing the same number of instructions after entering said partial speculation mode; and

returning to said full speculation mode after executing said same number of instructions.

18. The method of Claim 10 implemented using a microprocessor comprising a combination of translation software and host hardware, said translation software running directly on said host hardware, said translation software for interpreting and translating a sequence of non-native instructions into a sequence of native instructions, wherein said interpreting is permitted during said partial speculation mode.

19. A computer system comprising:

a main memory; and

a microprocessor coupled to said main memory;

wherein said computer system implements a first mode of speculative operation, a second mode of partial speculative operation, and a third mode in which speculative operations are suspended in entirety.

20. The computer system of Claim 19 wherein said first mode permits speculative operations comprising microprocessor register operations, operations that involve memory that is private to said microprocessor, input/output (I/O) writes, main memory reads, main memory writes, and non-architectural faults.

21. The computer system of Claim 19 wherein said second mode permits speculative operations comprising microprocessor register operations, operations that involve memory that is private to said microprocessor, and architectural faults.

22. The computer system of Claim 19 wherein said second mode permits speculative operations that are invisible external to said microprocessor.

23. The computer system of Claim 19 wherein a transition from said first mode to said second mode occurs in response to an event that is selected from the group consisting of a fault, a direct memory access request, and an I/O read.

24. The computer system of Claim 23 wherein a transition back to said first mode occurs after said event is handled.

25. The computer system of Claim 23 wherein the number of instructions executed in said first mode prior to said event are counted, wherein a transition back to said first mode occurs after the same number instructions are executed in said second mode.

26. The computer system of Claim 19 wherein said microprocessor is a microprocessor comprising a combination of translation software and host hardware, said translation software running directly on said host hardware, said translation software for interpreting and translating a sequence of non-native instructions into a sequence of native instructions, wherein said interpreting is permitted during said second mode.